Perancangan dan Implementasi API Master Sparepart Database menggunakan Framework Falcon (Studi Kasus : PT. Asuransi Sinar Mas)

Artikel Penelitian

Peneliti:
Derry Sandy Kurniawan (672015001)
Dr. Kristoko Dwi Hartomo, M.Kom.

Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Salatiga
Mei 2019

Perancangan dan Implementasi API Master Sparepart Database menggunakan Framework Falcon (Studi Kasus : PT. Asuransi Sinar Mas)

Artikel Penelitian

Diajukan kepada
Fakultas Teknologi Informasi
untuk memperoleh Gelar Sarjana Komputer

Peneliti:
Derry Sandy Kurniawan (672015001)
Dr. Kristoko Dwi Hartomo, M.Kom.

Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Salatiga
Mei 2019

PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama : DERRY SANDY KURNIAWAN

NIM : 672015001 Email : DERRYSANDYK@GMAIL.COM

Fakultas : TEKNOLOGI INFORMASI Program Studi : TEKNIK INFORMATIKA

Judul tugas akhir : PERANCANGAN DAN IMPLEMENTASI API MASTER SPAREPART DATABASE MENGGUNAKAN FRAMEWORK FALCON (STUDI KASUS: PT ASURANSI SINARMAS)

Pembimbing : 1. DR KRISTOKO DWI HARTOMO, M.KOM

2. _____

Dengan ini menyatakan bahwa:

1. Hasil karya yang saya serahkan ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar kesarjanaan baik di Universitas Kristen Satya Wacana maupun di institusi pendidikan lainnya.
2. Hasil karya saya ini bukan saduran/terjemahan melainkan merupakan gagasan, rumusan, dan hasil pelaksanaan penelitian/implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan narasumber penelitian.
3. Hasil karya saya ini merupakan hasil revisi terakhir setelah diujikan yang telah diketahui dan disetujui oleh pembimbing.
4. Dalam karya saya ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali yang digunakan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka.

Pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari terbukti ada penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya saya ini, serta sanksi lain yang sesuai dengan ketentuan yang berlaku di Universitas Kristen Satya Wacana.

Salatiga, 3 MEI 2019

DERRY SANDY KURNIAWAN

PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

PERNYATAAN PERSETUJUAN AKSES

Saya yang bertanda tangan di bawah ini:

Nama : DERRY SANDY KURNIAWAN
NIM : 672015001 Email : DERRYSANDYK@GMAIL.COM
Fakultas : TEKNOLOGI INFORMASI Program Studi : TEKNIK INFORMATIKA
Judul tugas akhir : PERANCANGAN DAN IMPLEMENTASI API MASTER SPAREPART DATABASE MENGGUNAKAN FRAWORK FALCON (STUDI KASUS: PT. ASURANSI SINAR MAS)

Dengan ini saya menyerahkan hak *non-eksklusif** kepada Perpustakaan Universitas – Universitas Kristen Satya Wacana untuk menyimpan, mengatur akses serta melakukan pengelolaan terhadap karya saya ini dengan mengacu pada ketentuan akses tugas akhir elektronik sebagai berikut (beri tanda pada kotak yang sesuai):

☑ a. Saya mengijinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA

☐ b. Saya tidak mengijinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA**

* *Hak yang tidak terbatas hanya bagi satu pihak saja. Pengajar, peneliti, dan mahasiswa yang menyerahkan hak non-ekslusif kepada Repositori Perpustakaan Universitas saat mengumpulkan hasil karya mereka masih memiliki hak copyright atas karya tersebut.*

** *Hanya akan menampilkan halaman judul dan abstrak. Pilihan ini harus dilampiri dengan penjelasan/ alasan tertulis dari pembimbing TA dan diketahui oleh pimpinan fakultas (dekan/kaprodi).*

Demikian pernyataan ini saya buat dengan sebenarnya.

Salatiga, 6 - MEI - 2019

DERRY SANDY KURNIAWAN
Tanda tangan & nama terang mahasiswa

Mengetahui,

Dr. KRISTOKO DWI HARTOMO, M.KOM.
Tanda tangan & nama terang pembimbing I

Tanda tangan & nama terang pembimbing II

FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN SATYA WACANA
Jalan Diponegoro 52 – 60
Phone. (0298) 321212 (Hunting)
Fax. (0298) 321433
E-mail: fti@uksw.edu
Salatiga 50711 – INDONESIA

LEMBAR PERSETUJUAN PUBLISH JURNAL

Dengan mempertimbangkan isi dari jurnal mahasiswa :

Nama Mahasiswa : DERRY SANDI KURNIAWAN
NIM : 672015001

Maka jurnal ini dinyatakan :

LAYAK TERBIT / TIDAK LAYAK TERBIT

Menyetujui,

Dr. KRISTOKO DWI HARTOMO, M.Kom
Pembimbing 1

Pembimbing 2

Mengetahui,

NAGOALENA A. WEKE PALEKENG, M.Kom
Reviewer

Lembar Pengesahan

Judul Tugas Akhir	: Perancangan dan Implementasi API Master Sparepart Database menggunakan Framework Falcon (Studi Kasus : PT. Asuransi Sinar Mas)
Nama Mahasiswa	: DERRY SANDY KURNIAWAN
NIM	: 672015001
Program Studi	: Teknik Informatika
Fakultas	: Teknologi Informasi

Menyetujui,

Dr. Kristoko Dwi Hartomo, M.Kom.

Pembimbing 1

Mengesahkan,

Dr. Wiwin Sulistyo, S.T., M.Kom.

Dekan

Magdalena A. Ineke Pakereng, M.Kom

Ketua Program Studi

Dinyatakan Lulus Tanggal: 02 Mei 2019

Reviewer :

- Magdalena A. Ineke Pakereng, M.Kom.

Perancangan dan Implementasi API Master Sparepart Database menggunakan Framework Falcon (Studi Kasus : PT. Asuransi Sinar Mas)

Oleh,

DERRY SANDY KURNIAWAN

672015001

ARTIKEL ILMIAH

Diajukan Kepada Program Studi Teknik Informatika Guna Memenuhi Sebagian Dari Persyaratan Untuk Mencapai Gelar Sarjana Komputer

Disetujui oleh,

Dr. Kristoko Dwi Hartomo, M.Kom.

Pembimbing 1

Diketahui oleh,

Dr.Wiwin Sulistyo, S.T., M.Kom.

Dekan

Magdalena A. Ineke Pakereng, M.Kom.

Ketua Program Studi

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS KRISTEN SATYA WACANA

SALATIGA

2019

Perancangan dan Implementasi API Master Sparepart Database menggunakan *Framework* Falcon (Studi Kasus : PT. Asuransi Sinar Mas)

Derry Sandy Kurniawan[1], Kristoko Dwi Hartomo[2]
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Jl. Dr. O. Notohamidjojo, Kel.Blotongan, Kec. Sidorejo, Salatiga 50714
Jawa Tengah - Indonesia
E-mail: 672015001@student.uksw.edu[1], kristoko@gmail.com[2]

Abstract

Master Spareparts Database is a database that serves to bring together the details of names and pictures of parts and prices from various sources. Master spareparts database are made from various sources such as the website of each ATPM, a desktop electronic catalog that has been put together and is ready for use but there is no information system that can handle the data. Technology that can be used is API. By implementing the API in Master Spareparts Database the database will facilitate the development of software applications outside the system or with different programming languages or platforms . The API that will be used using the Falcon. This research using the Falcon Framework, is feasible to use on many platforms or many users. It can be seen in the results of Baseline Testing and Normal Testing with the number of users 10 to 20 not failing to call the API. The scenario on the Stress Test also shows good results where user 1000 can make requests with a failure rate of 5%, so the API is very ready to be used in applications that exist in PT. Asuransi SinarMas . *Falcon Framework can assist in accessing Master Spareparts Database with many platforms or applications owned by* PT. Asuransi Sinar Mas *without significant API failure when making many requests.*

Key Words: *API, Falcon Framework, Framework.*

Abstrak

Master Sparepart Database adalah *database* yang berfungsi untuk menyatukan detail dari nama dan gambar suku cadang serta harga dari berbagai sumber. Master sparepart database yang telah dibuat dari berbagai sumber seperti situs *web* setiap ATPM, katalog elektronik desktop yang sudah disatukan dan siap digunakan akan tetapi belum adanya sistem yang dapat menangani data tersebut. Teknologi yang dapat digunakan adalah *API*. Dengan mengimplementasikan *API* dalam Master Sparepart database akan memudahkan pengembangan *software* di luar sistem maupun dengan bahasa pemrograman berbeda. *API* yang akan digunakan menggunakan *Framework* Falcon. Dalam penelitian ini dengan menggunakan Falcon, sudah layak digunakan pada banyak platform atau banyak *user*, Terlihat pada hasil pengujian *Baseline Testing* dan *Normal Testing* dengan jumlah *user* 10 sampai 20 tidak terjadinya kegagalan dalam memanggil API. Skenario pada *Stress Test* juga terlihat hasil yang baik dimana dengan *user* 1000 dapat melakukan *request* dengan tingkat kegagalan 5%, sehingga API sudah sangat siap digunakan pada aplikasi yang ada di PT. Asuransi SinarMas. Master Sparepart Database menggunakan *Framework* Falcon Studi kasus PT. Asuransi Sinar Mas dapat membantu dalam melakukan akses terhadap Master Sparepart Database dengan banyak *platform* atau aplikasi yang dimiliki oleh PT. Asuransi Sinar Mas tanpa terjadinya kegagalan yang signifikan pada *API* pada saat melakukan banyaknya *request* .

Kata Kunci*:* *API, Falcon Framework, Framework.*

---

[1] Mahasiswa Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana Salatiga.

[2] Staf Pengajar Fakultas Teknologi Informasi Universitas Kristen Satya Wacana Salatiga.

1. Pendahuluan

Saat ini JTUM bengkel dari PT. Asuransi Sinar Mas mengalami masalah dalam proses pengadaan suku cadang kendaraan dan dalam mengevaluasi keabsahan sebuah klaim. Dalam proses pengadaan suku cadang untuk penggantian komponen kendaraan, sering terjadi kesalahan dalam pemesanan suku cadang yang tepat. Hal ini disebabkan kurangnya informasi suku cadang yang dimiliki oleh surveyor dan supplier dikarenakan referensi informasi suku cadang tersebar pada beberapa tempat yaitu: situs web setiap ATPM, katalog elektronik desktop , dan buku katalog fisik. dari permasalahan tersebut diperlukannya database untuk menampung informasi di beberapa tempat tersebut.

Master Sparepart Database atau biasa disingkat MSD adalah database yang berfungsi untuk menyatukan detail dari nama dan gambar suku cadang serta harga dari berbagai sumber.Master sparepart database yang telah dibuat dari berbagai sumber seperti situs web setiap ATPM, katalog elektronik desktop yang sudah disatukan dan siap digunakan akan tetapi belum adanya sistem informasi yang dapat menangani data tersebut , sehingga dibutuhkannya sistem informasi mesin pencarian data *sparepart* untuk mengakses Master Sparepart Database.

Adapun salah satu solusi untuk mengatasi permasalahan dalam mengakses Master Sparepart database adalah dengan memanfaatkan teknologi yang ada sekarang ini. salah satu teknologi yang dapat diterapkan adalah penggunaan berupa teknologi penyimpanan data ini tidak memerlukan ruang yang besar dan dapat menghindari kerusakan data penting. Salah satu teknologi yang dapat digunakan adalah API. Pengimplementasian API dalam Master Sparepart database akan memudahkan pengembangan berbagai aplikasi perangkat lunak di luar sistem maupun dengan bahasa pemrograman atau *platform* baik itu desktop, website, mobile apps dapat mengakses layanan yang berada di *web service*. Selain itu penggunaan data seminimal mungkin dan kecepatan akses yang singkat merupakan tujuan lain dari penggunaan *web service* [1]. Metode RESTful memiliki keunggulan pada requested data yang lebih kecil dibandingkan metode SOAP. *Requested size* data yang lebih kecil memungkinkan *transmission time* yang lebih singkat, konsumsi daya lebih rendah dan *web service* yang lebih cepat dibanding dengan SOAP (Dudhe & Sherekar, 2014) [2]. API yang akan digunakan menggunakan *framework* API FALCON dengan bahasa pemrograman Python. Penggunaan API Falcon dengan alasan mempermudah dalam membangun sebuah HTTP API dan *backend apps*. dengan tujuan cepat ringan dan fleksibel [3].

2. Tinjauan Pustaka

"Kesulitan dalam mengintegrasikan data pada beberapa sistem yang berbeda menjadi salah satu masalah yang sering dialami oleh *developer*, mulai dari bahasa pemrograman, *Platform*, dan perangkat yang digunakan. Oleh sebab itu, perlu adanya pembuatan web *service* untuk sistem informasi akademik STT Terpadu Nurul Fikri dengan teknologi *REST*. Perancangan ini dibuat dua model API untuk dua modul yakni modul mahasiswa dan modul dosen. Pada modul ini akan menghasilkan data dengan format *JSON*. Teknik pengujian API menggunakan

teknik black box testing dengan tools aplikasi penguji Postman. Pengujian *API* dilakukan pada *prototype* aplikasi web *service*. Metode penelitian yang digunakan adalah *Unified Process* yang menggunakan kerangka kerja Yii *Framework* 2.0 " [6] . Perbedaan dari Penelitian tersebut yaitu penelitian sekarang menggunakan *framework* Falcon , sedangkan penelitian terdahulu menggunakan Framework YII. Perbandingan YII dan Falcon yaitu *framework* YII diperuntukkan untuk *Front-end* dan *Back-end* [4] sedangkan Falcon digunakan pada *Backend*, *framework* Falcon dikhususkan untuk pembuatan Web API sehingga mendapat performa yang lebih baik [5].

"Penyimpanan data di *smartphone* internal untuk permainan aplikasi yang bisa dimainkan oleh banyak pengguna tentu bukan sesuatu yang direkomendasikan. Berdasarkan pada Masalahnya, diperlukan aplikasi yang berguna sebagai penyedia data / pesan untuk game aplikasi. Maka layanan web adalah salah satu solusi, yang dapat berfungsi sebagai jembatan antara aplikasi dan *database*. Jadi dengan layanan web akan memudahkan beban pada aplikasi, karena pertukaran data telah disediakan oleh layanan web. Ini makalah berfokus pada implementasi layanan web sebagai media pertukaran data pada permainan aplikasi dan pengujian menggunakan aplikasi postman." [6]. Perbedaan dari penelitian yang akan dibahas ada pada *web service* tersebut tidak menggunakan *Framework* , Menurut artikel warungcode.com pada judul ini Keuntungan Menggunakan *Framework* yaitu *Framework* membuat program / sistem menjadi lebih mudah dan cepat karena *framework* telah menyediakan fungsi-fungsi yang dibutuhkan. Keuntungan menggunakan *framework* yakni penulisan kode kita akan memiliki standar [7].

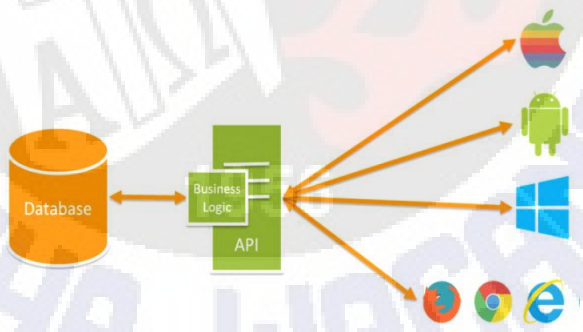
"Penelitian ini melakukan rancang bangun sistem pengaksesan nilai akademik berbasis desktop, dengan memanfaatkan web *services* sebagai penyedia data. Aplikasi yang dikembangkan nantinya dapat digunakan oleh mahasiswa sebagai aplikasi *student information center* (SIC) sehingga memudahkan dalam memperoleh data nilai akademik. Aplikasi yang dikembangkan dibangun menggunakan *library* nusoap dengan bahasa pemrograman PHP dan basis data MySQL, sedangkan pada sisi *client* yang mengkonsumsi web *services* menggunakan bahasa pemrograman microsoft visual basic .Net. Dari penelitian yang telah dilakukan terlihat bahwa web *services* dapat menjembatani pertukaran data dan informasi di lingkungan yang heterogen, baik di level sistem operasi komputer, sistem operasi jaringan, bahasa pemrograman dan basis data. Dengan adanya aplikasi ini, proses pengaksesan nilai akademik dapat dilakukan dengan lebih efektif." [8]. Penelitian Terdahulu menggunakan nusoap dimana arsitektur nya yaitu *SOAP* sedangkan penelitian sekarang menggunakan arsitektur *REST* perbedaan mendasar dari keduanya yaitu *SOAP* menggunakan penulisan *XML* dengan spesifikasi *SOAP* dan sifat nya tertutup hanya bisa diakses pada perusahaan atau vendor tertentu sedangkan *REST* menggunakan penulisan *JSON* dan sifat terbuka yang bisa diakses siapa saja [9].

A. *Framework*

Framework adalah sebuah kerangka kerja yang digunakan untuk mempermudah para *developer software* dalam membuat dan mengembangkan aplikasi. *Framework* berisikan perintah dan fungsi dasar yang umum digunakan untuk membangun sebuah software aplikasi sehingga diharapkan aplikasi dapat dibangun dengan lebih cepat serta tersusun dan terstruktur dengan cukup rapi. *Framework* juga bisa diartikan sebagai komponen-komponen pemrograman yang sudah jadi dan siap untuk digunakan kapan saja, sehingga pengembang aplikasi tidak perlu lagi membuat script yang sama untuk tugas-tugas yang sama [10].

B. *API (Application Program Interface)*

API adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. *API* terdiri dari berbagai elemen seperti *function, protocols, dan tools* lainnya yang memungkinkan *developers* untuk membuat aplikasi. Tujuan penggunaan *API* adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan *payment gateway*. Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk sistem operasi, *library*, dan web.



Gambar 1 Arsitektur API [10]

API yang bekerja pada tingkat sistem operasi membantu aplikasi berkomunikasi dengan *layer* dasar dan satu sama lain mengikuti serangkaian protokol dan spesifikasi. Web *API* seperti namanya diakses melalui protokol *HTTP*, ini adalah konsep bukan teknologi. Kita bisa membuat Web *API* dengan menggunakan teknologi yang berbeda seperti *PHP, Java, .NET*. Misalnya *Rest API* dari Twitter menyediakan akses *read* dan *write* data dengan mengintegrasikan *twitter* ke dalam aplikasi user sendiri [11].

Fitur dalam API [11] :

A. Mendukung fungsi *CRUD* yang bekerja melalui *HTTP protocol* dengan *method GET, POST, PUT* dan *DELETE*

B. Memiliki *response Accept Header* dan *HTTP status code*

C. *Response* dengan format *JSON, XML* atau format apapun yang kamu diinginkan. Akan tetapi kebanyakan digunakan ke dalam format *JSON*.

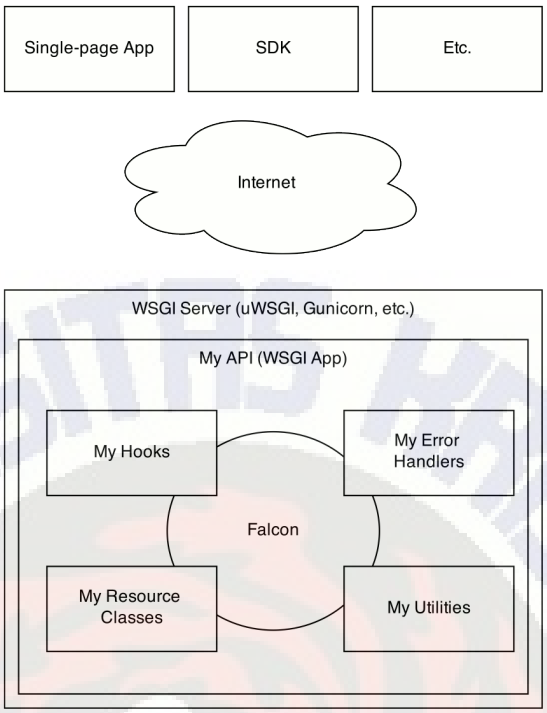
D. Mendukung fitur *MVC* seperti *routing, controllers, action results, filter, model, IOC container*.

E. Web *API* dapat berjalan di *Apache* atau *web server* lainnya yang didukung sesuai bahasa pemrograman yang digunakan.

Web *API* seperti sebuah alamat web (*end point*) yang dibuat untuk menangani beberapa *task* sesuai *request* yang diterima, juga terkadang memiliki *parameter* sebagai data yang dibutuhkan agar dapat menampilkan hasil yang diinginkan, juga pada beberapa kasus untuk mengakses *API* dibutuhkan kode otentikasi yang telah diizinkan untuk melihat data yang diinginkan. Semua *rule* ini ditentukan oleh *programmer* yang membuatnya [11].

C. Falcon Framework

Falcon Framework adalah *framework web* Python yang *reliable* dan *high-Performance* untuk membangun *app backends* yang responsif dan *microservices* berskala besar. Kelebihan utama *Framework* Falcon yaitu kecepatan dengan perangkat keras yang sama dan dengan lebih banyak permintaan. Falcon menyediakan *add-on, templates* serta *packages* yang siap digunakan untuk *project* yang akan dibuat. *Falcon* beberapa kali lebih cepat daripada kebanyakan *framework* Python lainnya. Untuk kecepatan tambahan, falcon mengkompilasi dengan Cython dan juga bekerja dengan baik dengan PyPy. Hasil Benchmarks falcon dengan 102,563 *req/sec* untuk Falcon Cythonized dan Falcon dengan 70,746 *req/sec* dibawah falcon ada beberapa framework lain seperti Bottle dengan 34,059 *req/sec*, Werkzeug dengan 9,948 *req/sec* Flask dengan 5,489 *req/sec* dan yang terakhir Django yang biasa dipakai untuk membuat web dari Python mendapat 3,712 *req/sec* di *score* terendah masing-masing menggunakan CPython 2.7.14 [12].

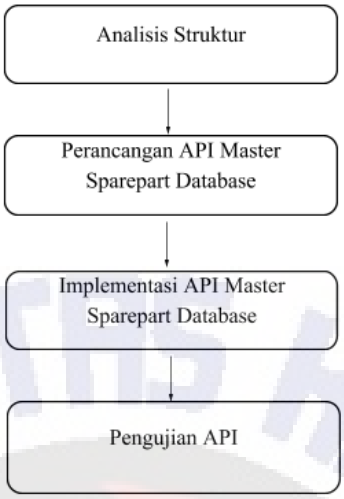


Gambar 2 Arsitektur Falcon [13]

Framework Falcon berjalan dengan arsitektur *REST*, terdiri dari dua bagian, yaitu *model, controller. Controller* berfungsi untuk menerima *request* dari *Client* juga untuk mengatur alur dari data. Model berfungsi untuk mengambil data dari database. *Web service* tidak menggunakan tampilan *User interface* karena berfokus pada layanan [1]. Falcon juga bekerja dengan *server* WSGI jenis apapun serta Falcon mendukung fitur ORM berjalan di Python versi 2.7 dan 3.3+ [13].

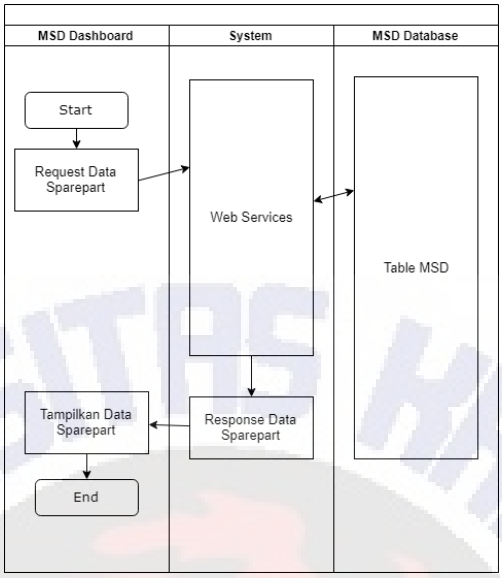
3. Metode Penelitian

Secara umum penelitian terbagi ke dalam 4 tahap, yaitu : (1) Analisis Struktur, (2) tahap Perancangan API Master Sparepart Database, (3) tahap Implementasi API Master Sparepart Database (4) tahap Pengujian dan Analisis Hasil Pengujian.



Gambar 3 Metodelogi Penelitian

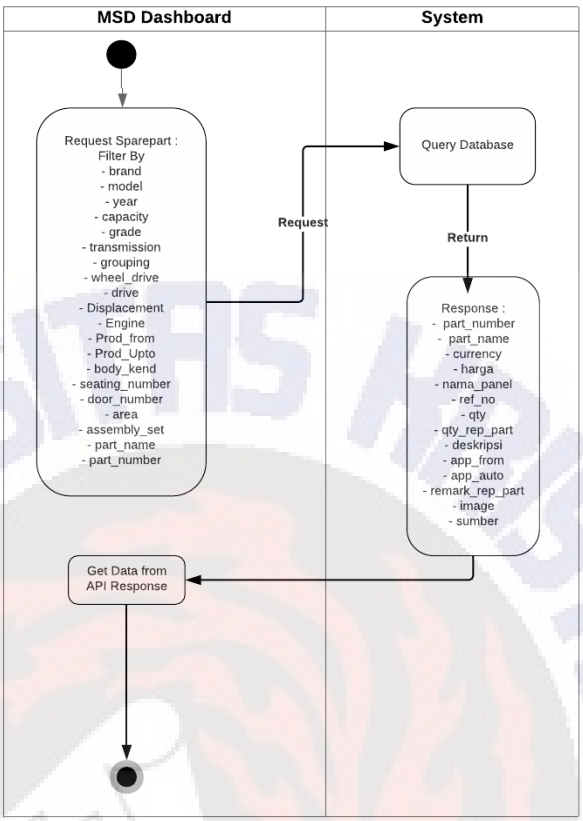
Tahapan penelitian pada Gambar 3 diawali dengan Tahap 1 yaitu Analisis Struktur, pada tahap ini dilakukan Mempelajari Struktur tabel yang diperlukan pada Master Sparepart Database API. Tahap 2 yaitu Perancangan API Master Sparepart Database, pada tahap ini Merancang API Master Sparepart Database dengan merancang *method* yang digunakan dan *response* agar sesuai dengan permintaan *user*. Tahap 3 yaitu Implementasi API Master Sparepart Database, pada tahap ini dilakukan Membuat API dengan menggunakan *Framework* Falcon di Python dengan rancangan di tahap Perancangan API Master Sparepart Database. Tahap 4 adalah Pengujian API. Pengujian dari API yang sudah dibuat apakah sudah sesuai dengan yang diharapkan *user* dan *front end* dengan implementasi ke sebuah sistem.



Gambar 4 Rancangan Keseluruhan API

Gambar 4 menunjukkan rancangan dari alur sistem yang akan dibuat yaitu :

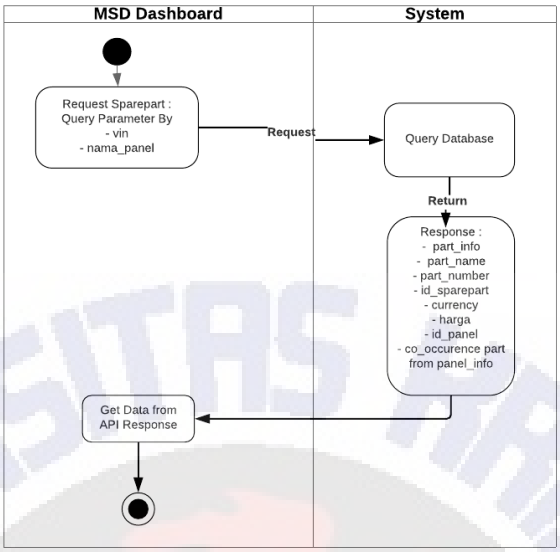
1. *Request data* : *request* data saat MSD Dashboard melakukan permintaan data ke *web services* yang akan dibuat.
2. *Web Service :* setelah menerima *request* dari aplikasi *web service* akan mengambil data sesuai permintaan dari aplikasi.
3. *Table* MSD : tabel MSD digunakan saat *web service* meminta data ke *database.*
4. *Response Data Sparepart* : setelah mengambil data sistem akan melakukan respon dari data yang diambil di *database* ke MSD *Dashboard*.



Gambar 5 Sparepart Search API

Gambar 5 menunjukan rancangan dari alur Sparepart Search API yang akan dibuat yaitu :

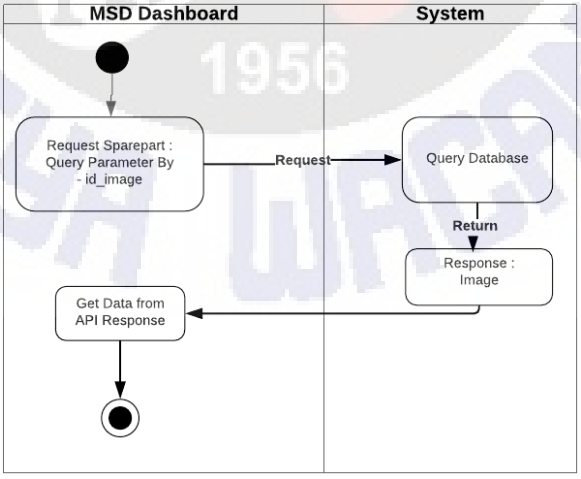
1. *Request* Sparepart Search : *request* Sparepart Search dengan seperti Gambar 5 dengan minimal satu parameter filter.
2. *Response* Sparepart Search : Respon dari Sparepart Search yaitu Data Sparepart di dalam filter sparepart.
3. Mengambil data *response* yang sudah di *request.*



Gambar 6 Panel Sparepart API

Gambar 6 menunjukan rancangan dari alur Panel *Sparepart* API yang akan dibuat yaitu :

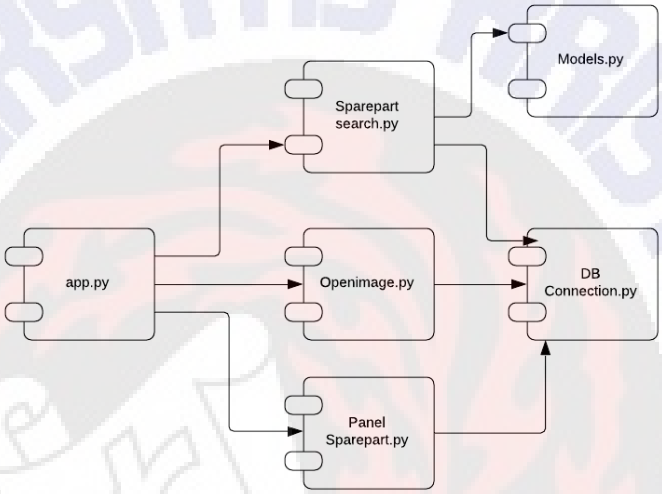
1. *Request* Panel *Sparepart* : *request* Panel *Sparepart* dengan seperti Gambar 3.4 .
2. *Response* Panel Sparepart: Respon dari Panel Sparepart yaitu data panel yang berkaitan dengan nama Panel
3. Mengambil data *response* yang sudah di *request*



Gambar 7 Open Image API

Gambar 7 menunjukan rancangan dari alur *Open Image API* yang akan dibuat yaitu

1. *Request* Panel *Sparepart* : *request Open Image*.
2. *Response* Panel Sparepart: Respon dari *Open Image* yaitu Gambar yang berkaitan dengan *Sparepart Search* API
3. Mengambil data *response* yang sudah di *request*



Gambar 8 Panel Sparepart API

Gambar 8 menunjukkan rancangan dari alur *Open Image API* yang akan dibuat yaitu :

1. *app.py* : menyatukan semua API yang akan digunakan serta melakukan *route* untuk setiap *endpoint* yang diperlukan.
2. *SparepartSearch.py*: API yang digunakan untuk mencari *Sparepart*.
3. *OpenImage.py* :API yang digunakan untuk mengambil data gambar pada data sparepart yang dicari.
4. PanelSparepart.py : API yang digunakan untuk mengambil data.
5. Models.py : file yang digunakan untuk Mapping table pada MSD *(Master Sparepart Database)* yang dipakai API *Sparepart Search*.
6. DBConnection.py : *file* yang digunakan untuk Membuat *Session Connection* pada *database*.

4. Hasil dan Pembahasan

Tabel 1

Tabel Daftar API Master Sparepart Database

Server	Nama API	Parameter	Method
MSD (Master Sparepart Database)	Panel *Sparepart*	- vin * - nama_panel *	GET
	Sparepart Search	- brand - model - year - capacity - grade - transmission - grouping - wheel_drive - drive - Displacement - Engine - Prod_from - Prod_upto - body_kend - seating_number - door_number - area - assembly_set - part_name - part_number	GET
	Open Image	- id_sparepart *	GET

(*) = Wajib Diisi

Tabel 1 merupakan tabel daftar API beserta parameter-parameter di setiap API dimana terdapat parameter wajib diisi dengan tanda (*) dan opsional. API yang terdapat pada *Server* MSD (Master Sparepart Database) dengan *method* yang digunakan GET . HTTP Server yang digunakan Gunicorn dijalankan melalui Docker dengan OS Environment Python 3.6.5 Slim Stretch pada Sistem operasi Centos 7.

Algoritma pada API yang sudah dibuat menggunakan *Falcon Framework* :

Algoritma 1 Sparepart *Search* API

1. Import falcon,json,sqlalchemy,session,models MsdTb* , alchemyencoder , aliased
2. Buat class sparepartsearchresource parameter object
3. Buat fungsi on_get parameter self,req,resp
4. Buat session
5. Panggil params

6. Ambil data params : brand, model, year, displacement, transmission, grouping, assembly_set, part_name, part_number, vin, grade, wheel_drive, drive, body_kend, seating_number, door_number, area, nama_panel
7. Buat alias dari semua model
8. Buat variabel fields berisi : id_sparepart , part_name,currency, harga , nama_panel, assembly_set , group_part, merk_kend, model_kend , prod_from , prod_upto , displacement, transmission , ref_no, qty, rep_part, qty_rep_part , deskripsi , app_from , app_auto , remark_rep_part , grade, wheel_drive , drive, body_kend , seating_number , door_number, area
9. Panggil session
10. Buat query dari fields distinct dari tabel vin dengan join Model_Covered, Sparepart, Harga, Alias_Panel, PanelMaster, Assembly_set, Grouping, ModelKend, ImageSparepart , LinkImageSparepart
11. Buat filter opsional
12. Grup 1
13. IF : Vin Ada : filter Vin
14. Grup 2
15. IF : Brand ada : Filter brand
16. IF : Model ada : Filter brand
17. IF : Year ada : Filter brand
18. IF : Displacement ada : Filter brand
19. IF : Transmission ada : Filter brand
20. IF : Grouping ada : Filter brand
21. IF : Assembly_set ada : Filter brand
22. IF : Grade ada : Filter brand
23. IF : Wheel_drive ada : Filter brand
24. IF : Drive ada : Filter brand
25. IF : Body_kend ada : Filter brand
26. IF : Seating_number ada : Filter brand
27. IF : Door_number ada : Filter brand
28. IF : Area ada : Filter brand
29. Penghubung grup 1 AND
30. Grup 3
31. IF : Part_name ada : Filter brand
32. IF : Part_number ada : Filter brand
33. IF : Nama_panel ada : Filter brand
34. Penghubung grup 2 OR
35. Rumus Filter : (Grup 1 AND Grup 2) and Grup3
36. Ambil data sparepart
37. Ambil id sparepart
38. Masukan data sparepart ke list sparepart
39. Panggil session
40. Buat query image dari tabel image_sparepart berdasarkan id_sparepart join linkimagesparepart, sparepart
41. Ambil data id_image
42. Masukan ke list link image
43. Masukan ke link image ke list_sparepart
44. Tampilkan sebagai json

Algoritma 1 merupakan algoritma pembuatan API sparepart Search dimulai dari melakukan Import library yang dibutuhkan serta membuat class dan fungsi sesuai dengan framework pada falcon, baris 4 - 9 melakukan inisialisasi library dan parameter yang sudah dibuat. Baris 10 - 36 melakukan pembuatan query dan pemanggilan sesuai yang sudah di rancangkan, pada baris 37 - 39 mengambil data sparepart. Baris 40 - 44 melakukan query pada tabel image berdasarkan id_sparepart, baris 45 memasukan data sparepart dan image yang sudah didapat sebagai json.

Algoritma 2. Panel Sparepart API

1. Import falcon,json,sqlalchemy,session,models MsdTb* , alchemyencoder , aliased
2. Buat class PanelInfo parameter object
3. Buat fungsi on_get parameter self,req,resp
4. Buat session
5. Panggil params
6. Ambil data params : vin_code , panel
7. Buat alias dari semua model
8. Buat variabel fields berisi : id_sparepart , part_name, part_number ,currency, harga , nama_panel, id Panel
9. Panggil session
10. Buat query panel utama dari fields distinct dari tabel vin dengan join Model_Covered, Sparepart, Harga, Alias_Panel, PanelMaster
11. Ambil data panel
12. Ambil id panel
13. Masukan data panel ke list data panel
14. Buat query panel_master , tampilkan indeks_lokasi join dengan tabel panel matching filter dengan panel dari get params
15. Masukan data panel_master ke list indeks_lokasi
16. Buat query panel match , tampilkan id_panel filter berdasarkan indeks_lokasi dalam list indeks_lokasi
17. Buat query coocurence dari fields dari tabel vin dengan join ModelCovered, Sparepart, Harga, AliasPanel, PanelMaster filter dengan vin ,tidak dalam variabel id_panel , id_panel dalam panel_match
18. Ambil data coocurence
19. Buat list Panel info
20. Panel info berisi data panel
21. Co_occurence berisi data coocurence
22. Tampilkan Panel Info sebagai json

Algoritma 2 merupakan algoritma pembuatan API panel info, sama seperti API Sparepart Search dimulai dari melakukan Import library yang dibutuhkan serta membuat class dan fungsi sesuai dengan framework pada falcon, baris 4 - 9 melakukan inisialisasi library dan parameter yang sudah dibuat. Baris 10 - 17 melakukan pembuatan query dan pemanggilan sesuai yang sudah di rancangkan, pada baris 18 mengambil data sparepart. baris 19 - 21 membuat list Panel Info berisi data panel dan list Co_occurence berisi data coocurence lalu ditampilkan sebagai json.

Algoritma 3. Open Image API

1. Import falcon,json,sqlalchemy,session,models MsdTb* , alchemyencoder , aliased
2. Buat class PanelInfo parameter object
3. Buat fungsi on_get parameter self,req,resp
4. Buat session
5. Panggil params
6. Ambil data params : id_image
7. Buat alias dari semua model
8. Buat variabel fields berisi : file_image
9. Panggil session
10. Buat query open_image dari fields distinct dari tabel sparepart dengan join link_image_sparepart , image_sparepart
11. Ambil data file_image bertipe blob
12. Baca data image sebagai gambar
13. Tampilkan gambar
14. buat content_type image/jpg

Algoritma 3 merupakan algoritma pembuatan API Open Image, sama seperti API Sebelumnya dimulai dari melakukan Import library yang dibutuhkan serta membuat class dan fungsi sesuai dengan framework pada falcon, baris 4 - 9 melakukan inisialisasi library dan parameter yang sudah dibuat. Baris 10 melakukan pembuatan query dan pemanggilan sesuai yang sudah di rancangkan, pada baris 11 mengambil data gambar. baris 12 - 13 membaca data gambar bertipe blob dengan fungsi pada falcon yaitu content_type:image/jpg dan menampilkannya.

Algoritma 4. Main APP

1. Import falcon
2. Import sparepart_search, Open_image, Panel_sparepart
3. Buat pemanggilan falcon API bernama app
4. Buat variabel sparepart_search , Open_image , Panel_sparepart
5. Isi variabel tersebut dengan nama class pada API
6. Menambah app.add_route /sparepart_search,/panel_sparepart,/image
7. Buat fungsi create()
8. Return app

Algoritma 4 merupakan algoritma pembuatan API untuk memanggil seluruh API yang dibutuhkan dalam MSD (Master Sparepart Database). baris 1 - 3 melakukan import file dan inisialisasi fungsi falcon, baris 4 - 5 melakukan inisialisasi API yang akan di gunakan. Baris 6 melakukan routing pada setiap API yang digunakan menggunakan fungsi .add_route , baris 7 - 8 memanggil fungsi yang nantinya akan dijalankan oleh server.

Pengujian Pertama akan dilakukan dengan menggunakan POSTMAN untuk mengetahui *Request* dan *Response* API yang sudah dirancang dan di implementasi. Tabel 2 hasil *black box testing* menggunakan POSTMAN

Tabel 2
Hasil *Black Box* Pada Pengujian Postman

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
1	Melakukan *request* pada sparepart search dengan 6 parameter kombinasi	Memunculkan hasil response sesuai yang di skenario dan hasil sesuai dalam parameter tersebut	Valid
2	Melakukan *request* pada Panel Sparepart	Memunculkan hasil response sesuai yang di skenario dan hasil sesuai dalam parameter tersebut	Valid
3	Melakukan request pada Open Image API dari response hasil yang diberikan sparepart search	Memunculkan Data Gambar	Valid

Request Sparepart Search dilakukan pada Tabel 2 dengan *method GET* . Contoh request menjadi seperti *"192.168.105.102:8000/sparepart_search/?brand=honda&model=mobilio&transmission=5mt&part_number=76841TG1T11&vin=MHRDD4850FJ4&part_name=tank"* Hasil dari request dapat dilihat pada Gambar 9.

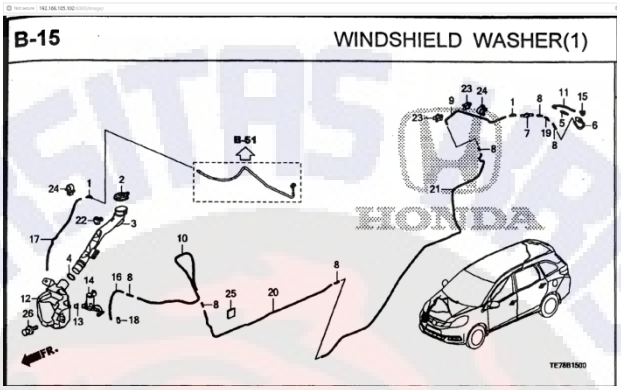
***Json response* 1**. Hasil Pencarian Sparepart Search

```
1.  "KATALOG": [
2.          {
3.              "id_sparepart": "170002023g002a0090008",
4.              "part_name": "TANK SET, FUEL ",
5.              "part_number": "17044TE8K00",
6.              "currency": null,
7.              "harga": null,
8.              "nama_panel": null,
9.              "ref_no": "1",
10.             "qty": "1",
11.             "rep_part": null,
12.             "qty_rep_part": null,
13.             "deskripsi": null,
14.             "app_from": 1500,
15.             "app_upto": null,
16.             "remark_rep_part": null,
17.             "image": [
18.                 "/image/?id_image=170002042a0380010001"
19.       "/image/?id_image=170002042a0380010002"
20.             ],
21.         },
```

Pada *Json response* 1 menunjukkan hasil *request* dari Sparepart Search dimana hasil response menampilkan satu data sparepart dari sumber baris 1 menunjukan sumber dari katalog dengan baris ke 2 - 22 menunjukan hasil sparepart dengan nama

sparepart *TANK, WASHER* dan lainnya yang memiliki dua gambar yang nantinya akan dipakai oleh API *Open Image.*

Parameter id_image yang didapat dari hasil API Sparepart Search pada Gambar 9. Contoh request OpenImage“http://192.168.105.102:8000/image/?id_image= 170002042a0380010001”. Hasil Gambar bisa dilihat pada Gambar 10



Gambar 10 Response Open Image

Gambar 10 menunjukkan gambar dari API Open Image dengan parameter id_image 170002042a0380010001 artinya di dalam gambar tersebut adanya sparepart dengan nama TANK, Washer.

Request panel sparepart pada tabel II dengan parameter vin dan nama panel yang ingin mencari panel terkait Contoh request panel sparepart “192.168.105.102:8000/vin_code/?vin=MHRDD4770EJ4&nama_panel=filter”. Hasil Gambar bisa dilihat pada Gambar 11.

***Json response* 2.** Hasil Pencarian Panel Sparepart


```
{
    "panel_info": [
        {
            "id_sparepart": "170002013g002a0376022",
            "part_name": "COVER, AIR CLEANER ",
            "part_number": "17210RE1Z01",
            "currency": null,
            "harga": null,
            "nama_panel": "FILTER UDARA",
            "id_panel": "30912"
        }
    ],
    "co_occurrence": [
        {
            "id_sparepart": "170002013g002a0011038",
            "part_name": "REE1 ASSY, CABLE(FURUKAWA)",
            "part_number": "77900TF0E11",
            "currency": null,
            "harga": null,
            "nama_panel": "KABEL SPIRAL",
            "id_panel": "30984"
        },
        {
```

```
24.             "id_sparepart": "170002013g002a0011038",
25.             "part_name": "REE1 ASSY, CABLE(FURUKAWA)",
26.             "part_number": "77900TF0E11",
27.             "currency": null,
28.             "harga": null,
29.             "nama_panel": "SENSOR TRANSMISI",
30.             "id_panel": "30619"
31.         },
```

Json response 2 baris 1 -12 menunjukkan data panel terdapat pada parameter vin dan nama panel yang terdapat pada panel_info. Baris 13 - 31 berisi *List* yang terdapat pada *co_ocurrence* adalah *list* panel yang terkait dengan baris 1 - 12.

Pengujian kedua menggunakan Locust aplikasi *Stress Test* pada Python dengan tiga skenario pengujian dapat dilihat pada Tabel 5.

Tabel 5
Skenario Stress Testing Pada API MSD
(Master Sparepart Database)

Test Name	***Min Wait***	***Max Wait***	***User Number***	***User Spawn Rate***	***Test Duration***
Baseline Testing	5000	15000	20	1	2 Hours
Normal Testing	10000	30000	10	1	8 Hours
Stress Testing	0	0	1000	100	1 Hours

Tabel 5 adalah Skenario *Stress Testing* dengan tiga skenario yaitu *Baseline, Normal, Stress Testing*.

Tabel 6
Hasil Stress Testing Pada Api Msd
(Master Sparepart Database)

Test Name	***Request***	***Fails***	***Request / Second***
Baseline Testing	43026	0	6
Normal Testing	38406	0	2
Stress Testing	667068	33876	184

Hasil *Baseline Testing* menunjukkan tiga *endpoint* yaitu */sparepart_search* , */vin_code, /image* dengan parameter pada *Json response* 4 telah melakukan *request* sebanyak 43026 dengan kegagalan *request* 0 dengan *request/second* 6 selama 2 jam dengan jumlah pengguna 20. Hasil *Baseline Testing* bisa dikatakan berhasil karena tidak ditemukan nya kegagalan/*fails* pada hasil *Baseline Testing*.

Hasil Normal *Testing* telah melakukan *request* sebanyak 38406 dengan kegagalan *request* 0 dengan *request/second* 2 selama 8 jam dengan jumlah pengguna 10. Hasil Normal *Testing* bisa dikatakan berhasil karena tidak ditemukan nya kegagalan/*fails* pada hasil Normal *Testing*.

Hasil Stress *Testing* telah melakukan *request* sebanyak 667068 dengan kegagalan *request* 33876 dengan *request/second* 184 selama 1 jam dengan jumlah pengguna 1000. Hasil *Stress Testing* bisa dikatakan berhasil karena dari total request hanya mengalami kegagalan 33876 dari 667058 jika diubah kedalam persen menjadi 5% dari total keseluruhan 100%.

5. Simpulan

Berdasarkan hasil Pengujian menggunakan Locust, maka dapat diambil kesimpulan bahwa API *Framework* Falcon yang dibuat sudah layak digunakan pada banyak *platform* atau banyak *user*, Terlihat pada hasil pengujian *Baseline Testing* dan *Normal Testing* dengan jumlah *user* 10 sampai 20 tidak terjadinya kegagalan dalam memanggil API. Skenario pada *Stress Test* juga terlihat hasil yang baik dimana dengan *user* 1000 dapat melakukan *request* dengan tingkat kegagalan 5%, sehingga API sudah sangat siap digunakan pada aplikasi-aplikasi yang ada di PT. Asuransi Sinar Mas. Penerapan API dalam aplikasi terdapat pada Tabel 1 dengan parameter dan kondisi yang dibutuhkan pengguna.

Berdasarkan hasil penelitian dan pengujian maka dapat disimpulkan bahwa Perancangan dan Implementasi API Master Sparepart Database menggunakan Framework Falcon Studi kasus PT. Asuransi Sinar Mas dapat membantu dalam melakukan akses terhadap Master Sparepart Database dengan banyak platform atau aplikasi yang dimiliki oleh PT. Asuransi Sinar Mas tanpa terjadinya kegagalan yang signifikan pada API pada saat melakukan banyaknya request .

6. Daftar Pustaka

[1] Rohman, R.F , Dkk. (2017). Pengembangan Perangkat Lunak Aplikasi Monitoring Klimatologi Menggunakan Metode Restful Web Service Berbasis Android (Studi Kasus : Stasiun Klimatologi Karangploso Malang) . Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, Vol. 2, No. 6

[2] Dudhe, A. & Sherekar, S., 2014. Performance Analysis Of Soap And Restful Mobile Web Services In Cloud Environment. Ijca Special Issue On Recent Trends In Information Security, Vol Rtinfosec, Pp. 1-4

[3] Dian , Muhar, 2014. Belajar Pemrograman Python : Pengenalan Dasar Python Dan Persiapan Awal Https://Www.Petanikode.Com/Python-Linux/ Diakses Pada 9 Juli 2018.
[4] Arianto, M.A., Munir,S., & Khotimah, Khusnul. (2016). Analisis Dan Perancangan Representational State Transfer (Rest) Web Service Sistem Informasi Akademik Stt Terpadu Nurul Fikri Menggunakan Yii Framework. Jurnal Teknologi Terpadu , Vol. 2, No. 2
[5] Yii Framework, Desember 2004. The Definitive Guide To Yii 2.0 Https://Www.Yiiframework.Com/Doc/Guide/1.1/Id/Quickstart.What-Is-Yii Diakses Pada 16 Juli 2018
[6] Zaman, G.A.P. (2017). Perancangan Dan Implementasi Web Service Sebagai Media Pertukaran Data Pada Aplikasi Permainan . Jurnal Informatika Vol. 11, No. 2
[7] Warungcode, Desember 2017. Ini Dia Keuntungan Menggunakan Framework Https://Warungcode.Com/Ini-Dia-Keuntungan-Menggunakan-Framework/ Diakses Pada 13 Juli 2018.
[8] Amin, M.M. (2017). Pengembangan Layanan Akses Nilai Akademik Berbasis Web Services . Jurnal Jupiter, Vol. 9 No. 1 , Hal. 13 - 22
[9] Nurdiyanto W, Desember 2012. Perbandingan Soap Dan Rest Sebagai Web Service Http://Pusdiklat.Bps.Go.Id/Index.Php?R=Artikel/View&Id=206 Diakses Pada 16 Juli 2018.
[10] Utopicomputer.Com, Februari 2014. Apa Itu Framework ? Berikut Pengertian Dan Fungsinya Https://Www.Utopicomputers.Com/Apa-Itu-Framework-Berikut-Pengertian-Dan-Fungsinya/ Diakses Pada 9 Juli 2018.
[11] Sandi, A., November 2017. Mengenal Apa Itu Web Api Https://Www.Codepolitan.Com/Mengenal-Apa-Itu-Web-Api-5a0c2855799c8 Diakses Pada 9 Juli 2018.
[12] Falcon, Februari 2012. Falcon Introduction Https://Falconframework.Org Diakses Pada 12 Juli 2018.
[13] Falcon, Februari 2012. Falcon Introduction Https://Falcon.Readthedocs.Io/En/Stable/User/Intro.Html Diakses Pada 9 Juli 2018.
[14] Yusrizal., Dawood , R., & Roslidar (2017). Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter . Jurnal Online Teknik Elektro , Vol.2 No.1 , Hal 1-8
[15] Krenov K. , Juni 2016. Python's Web Framework Benchmarks Http://Klen.Github.Io/Py-Frameworks-Bench/ Diakses Pada 24 Juli 2018.
[16] Dian , Muhar, 2014. Belajar Pemrograman Python : Pengenalan Dasar Python Dan Persiapan Awal Https://Www.Petanikode.Com/Python-Linux/ Diakses Pada 9 Juli 2018.